Spark Detection

for MuCool project

Fri, Sep 6, 2002

A spark-influenced vacuum waveform is sampled at 15 Hz. When a spark occurs, this waveform rises in about a second and falls in several seconds. Its amplitude may be about 3–4% of a signal size of 3.5 volts. This note describes the local application SPRK that uses logic to detect and count sparks exhibiting such behavior.

Included with the averaging code is logic that is used for two separate vacuum readings that detects when excursions above a specified threshold are found. Monitoring logic observes times when both vacuum readings are seen to be above the threshold. If such times occur, but for not too long, a spark is counted, and recent captured vacuum readings are stored in a data stream record as evidence that can be examined to build confidence in what the program counts as a spark.

The above brief description shows that many parameters must be chosen to fully characterize the program's exact behavior. As a local application, several parameters are used, as follows:

```
Param       Value   Meaning
ENABLE  B    00D0   Enable Bit#
SPARKS  C    00B0   Spark counter channel#
VAC1    C    0110   Vacuum#1 channel#
THR PCT1        2   Percent Vac1 signal for spark
VAC2    C    0111   Vacuum#2 channel#
THR PCT2        2   Percent Vac2 signal for spark
--
--
--
TARGNODE     0584   Target node used for testing
```

The TARGNODE parameter facilitates testing of the program by allowing one to specify a node# from which the vacuum data should be read. This parameter will normally be zero, meaning that the data will be obtained from the local node. The values shown here are the ones used for initial testing of the program in node0509. The threshold percentages signify at what percent above the average value a spark is first recognized. These values are only set here as a whole number of percent. (The program could be modified to take these values from dummy channels, which may be the easiest way to permit using fractional percentage values.) The spark counter channel is one whose reading is incremented for each spark detected. After the count reaches 32767, no further incrementing is possible. One may clear this spark count by setting a zero value to this channel. The vacuum channel numbers hold the two vacuum readings of interest. If it were necessary to operate the program with only a single vacuum channel, use the same channel number in both parameters. Any of these parameters can be modified in the usual way, via Page E or its equivalent. But to make the changes effective, the program logic should be restarted by toggling the enable bit.

In addition to the above parameters, several others are specified as program constants. These can be altered by merely recompiling and downloading to the node using the program.

```
Param  Value   Meaning
nAvg     100   #cycles of 15 Hz readings for averaging and history
nWave     60   #cycles of 15 Hz readings saved in data stream record
overMax    3   #cycles to confirm above threshold
```

```
underMax    3   #cycles to confirm below threshold
bothMax    40   #cycles after which spark no longer recognized
futureMax  20   #cycles of data to include after spark is detected
```

When the program starts up, it initializes its variables and incorporates the parameter values specified in the first list. It first builds up the history/averaging array, which requires `nAvg` cycles. By this time it has a valid average value to use as a reference for spark detection. On each subsequent cycle, for each vacuum reading, it revises the average value, computes the threshold using the percent parameter, and checks whether the amount by which the current vacuum reading exceeds the average has exceeded the threshold. If it has, it awaits `overMax` cycles to confirm, then declares a provisional spark status as seen by that vacuum reading.

After performing the above logic for each of the vacuum readings, it ANDs the results, looking for both to indicate the above threshold state. If that combined state lasts for not more than `bothMax` cycles, a spark is officially recognized. After that, `futureMax` number of additional readings are collected, the spark channel is incremented, and a record is written to a data stream called "`SPRKLOG` ", if one exists. (During the additional collection period, the logic is not looking for sparks, but the averaging logic continues apace.) The record written to the data stream includes the current date and time, the count of such records written, the number of cycles both readings were found to be above the threshold, and the last `nWave` points of each vacuum reading. These waveforms can be examined to check whether a human's idea of a spark matches the computer's idea of a spark.